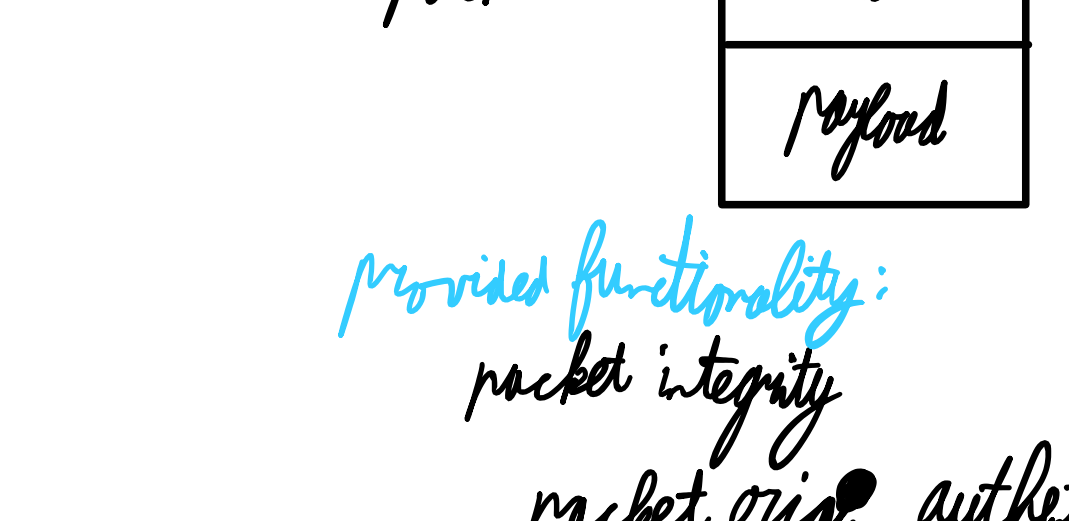


Secure communication



- application: mess changes every application... **PGP, S/MIME, OTN** → only gives point-to-point security
- transport: encrypt sessions/messages **client-server** → **TLS/SSL** → no encryption between front-end and application servers
- network: encrypt IP packets **IPSec, Wireguard**
- link: data security between endpoints → **WPA2**
- Physical layer



- provides functionality:
- packet integrity
 - packet origin authentication
 - confidentiality
 - traffic flow confidentiality
 - replay attack protection

transport mode: only payload is protected, header is not encrypted; ESP, parts of header and entire payload are authenticated
 only used between hosts

tunnel mode: header and data are protected; put (encrypted) into new packet; provides data flow confidentiality (to some extent)
 typically used for VPN

- authentication header: (AH)
- next header (type)
 - payload length (of AH)
 - security parameter index (SPI)
 - sequence number: against replay attacks
 - authentication data (MAC)

- ESP packet contents
- SPI
 - sequence number
 - payload data
 - padding
 - padding length
 - next header
 - authentication data

- security association: parameters
- sequence number, overflow
 - anti-replay window
 - ⋮

Security policy database: describes what security associations are supported by certain ports

Should be both encryption & authentication

many key - agreement protocols exist

key exchange security requires

- confidentiality
- authenticity: of at least one party
- correctness
- forward secrecy: confidentiality remains if one party's keys are compromised
- post-compromise security: re-establish security after compromise

AE = authenticated encryption
 AD = associated data

KEM = key encapsulation method
 AKE = authenticated key exchange
 FS-AKE = forward-secure AKE

↳ add forward key: is that a diffie, when key is used for every session; key forward key used 'usual' key
 ↓
 since this is destroyed after session, recovery of plaintext becomes impossible after session

es → indicator has ephemeral key
 responder has static key

h = handshake hash: hash of all handshake messages using hash
 CK = chaining key: function of all DH results so far using HKDF

new CK ⇒ derive session key k & nonce n
 ↓
 encrypt payload with k

N = no static key
 K = known
 X = transmitted
 I = immediately transmitted

goals: authenticity: none, no KCI, KCI
 KCI = key compromise impersonation

confidentiality: none, different levels of forward secrecy

identity binding

Wireguard: simple VPN protocol
 noise with IK pattern

TLS = variant of SSL v3
 ↳ designed for web, also used for mail/SMTP...
 ↳ provides transparency for higher (i.e. application) layers
 ↳ runs on top of TCP

